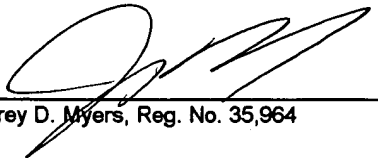




**PATENT APPLICATION**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Box AF: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on June 11, 2004.

  
Jeffrey D. Myers, Reg. No. 35,964

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: John N. Kesler

Examiner: J. Veillard

Serial No. 10/040,851

Group Art Unit: 2175

Filed: December 28, 2001

For: AUTOMATED GENERATION OF DYNAMIC  
DATA ENTRY USER INTERFACE FOR  
RELATIONAL DATABASE

SECOND DECLARATION OF JOHN N. KESLER UNDER 37 C.F.R. § 1.68 and 1.131

Commissioner for Patents  
P.O. Box 1450  
Alexandria, Virginia 22313-1450

Sir:

John N. Kesler declares as follows:

1. I am the inventor of the subject matter of the above-identified patent application.
2. My invention was actually reduced to practice as prototype software prior to March 16, 2001.
3. Attached as Exhibit A to my first Declaration, mailed January 28, 2004, is a program listing of CD-ROM copy of the backup source code of the prototype software as it existed in about November 2000. The software was then known as AppSwift and shortly thereafter as AppQuest. In January 2004, I

reviewed the source code listed in Exhibit A. I further reviewed the source code in June 2004 to prepare this second Declaration.

4. My review verified that the prototype software as of November 2000 had the following characteristics and capabilities as claimed in claims 1, 8, and 15 of the present patent application: The software was computer software for automatically generating a user interface for a relational database, the software comprising: utility software extracting schema information from the relational database and automatically generating corresponding schema and user interface metadata; a repository for the metadata; and user interface software automatically developing from the metadata a user interface appropriate to the relational database. The software was executable on a computer and thereby formed a computer apparatus for automatically generating a user interface for a relational database, the apparatus comprising: means for extracting schema information from the relational database and automatically generating corresponding schema and user interface metadata; repository means for containing the metadata; and means for automatically developing from the metadata a user interface appropriate to the relational database. The software also provided a method of automatically generating a user interface for a relational database, the method comprising the steps of: extracting schema information from the relational database and automatically generating corresponding schema and user interface metadata; storing the metadata in a repository; and automatically developing from the metadata a user interface appropriate to the relational database.

5. Furthermore, in February 2001 the brochure attached as Exhibit B to my first declaration was developed. It accurately describes the capabilities of the software at that time. Exhibit C to my first declaration is a check that was issued on February 22, 2001 to the company that developed the brochure. This check was issued after work by the company was complete.

6. Further supporting documentation is now provided in response to the Office Action dated April 14, 2004. The supporting documentation maps the elements of independent claims 1, 8, and 15 of the present patent application with the source code that existed in November of 2000. The organization of the supporting documentation is summarized as follows:

#	Claim Recital	Patent Application Section	Exhibit
1	"a repository for said metadata"	The Configuration Repository (pages 9 through 13)	D
2	"utility software extracting schema information from the relational database and automatically generating corresponding schema and user interface metadata"	Metadata Generation (pages 18 through 22)	E
3	"user interface software automatically developing from the metadata a user interface appropriate to the relational database"	User Interface Generation (pages 22 through 39)	F

**Claim Element - "a repository for said metadata"**

The repository metadata is itself a relational database implemented using Microsoft Access. The metadata database was located in the following file listed on Exhibit A to my first declaration:

"AdministrativeTool\SystemInfo.dat".

Entity relationship diagrams (ERD) for the portions of the metadata database listed in Exhibit A to my first declaration and discussed in the Configuration Repository section of the patent application are given in Exhibit D hereto. A table of contents for Exhibit D is included below. Please note that the metadata components listed in the table are limited to those impacted by the metadata generation process. The configuration repository includes other metadata which supports additional functionality such those items discussed in the Application Extensibility section of the patent application:

<b>Metadata Component</b>	<b>Page</b>
Entities	1
Entity Fields	2
Entity Relationships	3
Entity Search Path	4
Entity and Entity Relationship Permissions	5
Platform and Platform Attributes	6

**Claim Element – “utility software extracting schema information from the relational database and automatically generating corresponding schema and user interface metadata”**

The metadata generation process described in the Metadata Generation section of the patent application is initiated by a system administrator selecting the Refresh menu option of the Utility Tool (after having created a new repository file). This action results in a series of function calls that populate the

configuration repository with database schema and user interface metadata. This metadata is then utilized by the User Interface to generate the end user interface.

After the initial population of the metadata, the Refresh process is repeated when changes are made to the underlying relational database structure. Further configuration changes to the metadata are made possible through a series of data entry screens accessible through the Utility Tool's main screen located in "AdminstrativeToolMain.frm" listed in Exhibit A to my first declaration.

The following table contains an ordered list of each of the function calls made during the Refresh process. The CD File Location column indicates the physical file listed on Exhibit A to my first declaration in which each function is located. The relevant source code text is provided in Exhibit E hereto. Indentation is used to denote function calls within functions:

<b>Function</b>	<b>File Location</b>	<b>Page</b>
Refresh Menu Option	AdminstrativeToolMain.frm	1
RefreshDatasource	AdminstrativeTool\General.bas	2
VerifySequenceTable	AdminstrativeTool\General.bas	5
VerifyDualTable	AdminstrativeTool\General.bas	6
InsertDummyRecord	AdminstrativeTool\General.bas	7
VerifyAuditTable	AdminstrativeTool\General.bas	8
GetPrimaryKey	AdminstrativeTool\General.bas	9
GetRecordsource	AdminstrativeTool\General.bas	10
IsReferenceTables	AdminstrativeTool\General.bas	11
CPermission.CreatePermissions	AdminstrativeTool\CPermission.cls	12

ModifyRelationships	AdminstrativeTool\General.bas	13
OrderEntities	AdminstrativeTool\General.bas	14
CreateRelationships	AdminstrativeTool\General.bas	15
RefreshRelationships	AdminstrativeTool\General.bas	16
AddRelationship	AdminstrativeTool\General.bas	17
RefreshEntities	AdminstrativeTool\General.bas	18
RefreshEntity	AdminstrativeTool\General.bas	19
CreateComboBoxControls	AdminstrativeTool\General.bas	21
BuildRecordsource	AdminstrativeTool\General.bas	22
CreateComboBoxControl	AdminstrativeTool\General.bas	23
CreateSmartComboControl	AdminstrativeTool\General.bas	24
OrderEntityColumns	AdminstrativeTool\General.bas	25
OrderRelationships	AdminstrativeTool\General.bas	26
LoadSequences	AdminstrativeTool\General.bas	27
InsertSequence	AdminstrativeTool\General.bas	28
RefreshContexts	AdminstrativeTool\General.bas	29
DefineContexts	AdminstrativeTool\General.bas	30
DefineContext	AdminstrativeTool\General.bas	31

**Claim Element - “user interface software automatically developing from the metadata a user interface appropriate to the relational database”**

The user interface software, located under the "UserInterface" directory, is comprised of six components each of which is an ActiveX DLL, an ActiveX OCX, or an executable program (EXE). The primary function and location of each component are summarized in the following table:

<b>Component</b>	<b>Directory Location</b>	<b>Primary Function</b>
AppUtils.dll	UserInterface\AppUtils	Contains enumerated types and global functions.
AppInfo.dll	UserInterface\AppInfo	Data structure for metadata. Contains methods for retrieval of all metadata. Responsible for loading metadata.
Framework.dll	UserInterface\Framework	Implements non-GUI logic such as all database interaction.
QueryTool.ocx	UserInterface\QueryTool	An end user database query tool not discussed in the patent application.
ReportTool.ocx	UserInterface\ReportTool	Provides for integration of hardcopy reports. Briefly discussed under the Application Extensibility section of the patent application.
AppSwift.exe	UserInterface\Interface	The main user interface which is comprised of the previous five components. Provides all screens through which end users interact with the relational database.

The User Interface Generation section of the patent application discusses the methodology employed to generate the user interface from the metadata. Given that the user interface is implemented using object-oriented and event driven programming techniques, a complete procedural listing of the source code in Exhibit F hereto would be impractical. However, the following table can be used as a guide

to the source code listed in Exhibit F (by page) corresponding to the specific topics discussed in the User Interface Generation section of the patent application:

<b>Patent Topic</b>	<b>CD File Location</b>	<b>Page</b>
View Menu	UserInterface\Interface\Source\GenericForms\Main.frm	1
Parent Entities	UserInterface\Interface\Source\GenericControls\EntityView.ctl	3
Quick-Text Search	UserInterface\Interface\Source\GenericControls\EntityView.ctl	5
Advanced Searching	UserInterface\Interface\Source\GenericForms\GenericFilter.frm	8
Child Entities	UserInterface\Interface\Source\GenericControls\Relations.ctl	9
Menus and Menu Contexts	UserInterface\Interface\Source\GenericControls\ListBoxControl.ctl	11
Data Entry Forms	UserInterface\Interface\Source\GenericForms\Input.frm	13
Data Entry Controls	UserInterface\Interface\Source\GenericControls\DataEntryControl.ctl	21
Entity Business Rules	UserInterface\Framework\clsFrameworkUtils.cls	33
Entity Triggers	UserInterface\Framework\clsFrameworkUtils.cls	35

7. I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the above-referenced application or any patent issuing thereon.



Date: 6-11-2004

  
\_\_\_\_\_  
John N. Kesler

G:\AMDS\AppQuest\AppQuest\_2d\_DCL.doc